

## **REMARKS/ARGUMENTS**

### **Overview of the Office Action**

Claims 1-13, and 15-29 were rejected under 35 U.S.C. § 102(e) as being anticipated by Chan et al. (U.S. Patent No. 5,893,107).

Claim 14 was rejected under 35 U.S.C. § 103(a) as being obvious over Chan in view of MSDN ("Lowering Total Cost of Ownership with Active Directory-Enabled Applications").

### **Status of the Claims**

Claims 1, 6, 15, 18, and 24 have been amended. Claims 1-29 are pending.

### **Explanation of Claim Amendments**

Claims 1, 15, 18, and 24 have been amended to reverse and effectively withdraw the amendments previously made by Applicant to said claims in order to return said claims to their original as-filed form.

For future listings of these claims, Applicant is unclear as to whether said claims should be annotated as "previously presented" or "original." Applicant would appreciate any clarification the Examiner could provide in this regard; however, absent any clarification from the Examiner, Applicant will hereafter indicated in any subsequent listing that said claims are "original" (unless and until subsequent amendments are made to said claims) to reflect that these claims are once again presented for examination in their original as-filed form.

**Claims Rejected Under 35 U.S.C. § 102(e)**

Claims 1-13 and 15-29 were rejected under 35 U.S.C. § 102(e) as being anticipated by Chan et al. (U.S. Patent No. 5,893,107). However, Applicant respectfully submits that Chan fails to teach each and every element present in Claims 1-13 and 15-29. Specifically, Chan fails to teach the use of an “aggregatable software object” as explicitly set forth in Applicant’s claims.

As known and appreciated by those of skill in the art, and as explicitly set forth in the Specification of the present Application, an “aggregatable object...is an object that can be aggregated to another object” where “aggregation is a specialized form of containment...which allows an internal object’s interfaces to be exposed as interfaces of an external object” (Application, page 14, lines 13-17). Significantly, it is the interfaces of the aggregatable object that are “added to the directory services interfaces [sic] to provide the directory services interface with the extended functionality” provided by said aggregatable object (Application, page 14, lines 6-8). This approach, illustrated in Fig. 7 of the present Application, is described as follows:

The directory object 700 [a directory services interface] has aggregated therein the extension object 702 [an aggregatable object]. Thus, the interfaces 704 that are normally a part of the [directory] object 700 are now extended to include the interfaces 706 of the extension object 702 [the aggregatable object]. Thus, **to a client of the directory object 700, both the interfaces 704 and the extended interfaces 706 are exposed.**

(Application, page 16, lines 12-16). Thus it is the interfaces of the extension object (which is an aggregatable object) that are directly extended by the directory service interface to a client, and a client can directly use the extension services by querying said extension service interfaces exposed by said directory service interface.

In comparison, the invention of Chan is directed to “provid[ing] a directory service system for accessing a plurality of directory services in a uniform manner” where “[e]ach directory service manages information relating to objects or that directory service” (Chan, Abstract, lines 1-4). The invention of Chan further comprises an “extending component” that “controls the defining of new object classes and new properties for each directory service” where “[a] client of the directory service system uses the extending component to define new object classes and new properties” (Chan, Abstract, lines 24-28). However, the invention of Chan does not enable its “directory service system” (corresponding to “directory services interface” of the present invention) to directly expose the interfaces of its “extending components” (corresponding to the “extension objects” of the present invention) to clients, and thus said extending components cannot be said to be “aggregatable objects” because the interfaces of these extending components are not being exposed to any client by the directory service system (which is also referred to as the “OleDs”; see Chan, col. 6, lines 8-13). Instead, the directory service system of Chan uses a “pre-defined” set of APIs to access extending components that have been mapped to said predefined set of APIs. More specifically:

...[T]he directory service system, referred to as OleDs, provides an architecture that **allows clients to access the contents of these various directory services using a single, common set of OleDs objects, attributes, and interfaces**. An OleDs object is an in-memory data structure that corresponds to an object of a directory service or to an object class of the directory service. When an OleDs object corresponds to an object, it provides an interface for accessing the property values and methods of that object. When an OleDs object corresponds to an object class, it provides an interface for accessing the definition of that object class. **Each provider of a directory service provides an implementation of these interfaces for their directory service that maps the behavior of their API set to the behavior of these**

**OleDs interfaces.** In this way, a client that is developed to use the OleDs architecture can access each of these directory services, **regardless of the differences in their API sets.**

(Chan, col. 6, lines 9-25). From this section, it is therefore apparent that the directory service system of Chan uses a single, common set of interfaces that must be utilized by the provider of a directory service to map the behavior of this standard interface set to the functionality of the service in order to overcome the difference between the predefined interfaces and the interfaces of the extending components.

Because the extending components of the Chan invention do not have their interfaces directly exposed by the directory service system, these extending components cannot be said to be “aggregatable objects” as this term is explicitly defined in the present Application. Since Claims 1-13 and 15-29 are all directed and limited to the utilization of aggregatable objects (which, by definition, have their interfaces directly exposed to clients by the directory service interface), and since Chan fails to teach the utilization of aggregatable objects, Applicant respectfully submits that Claims 1-13 and 15-29, Applicant respectfully submits that Chan fails to teach all the claim elements necessary to anticipate the invention of Claims 1-13 and 15-29 under 35 U.S.C. § 102(e). Therefore, Applicant respectfully requests that the rejection of Claims 1-13 and 15-29 be withdrawn and that these claims be allowed to issue.

**Claims Rejected Under 35 U.S.C. § 103(a)**

In order to establish a prima facie case of obviousness, three basic criteria must be met. First there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second there must be a reasonable expectation of success. Finally

the prior art reference (or references when combined) must teach or suggest all the claim elements. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art and cannot be based on applicant's disclosure. (MPEP §§ 2142, 2143.)

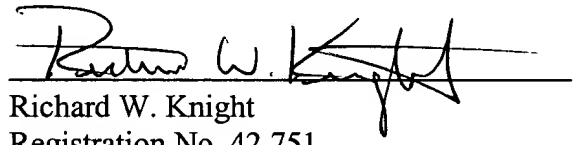
Claim 14 was rejected under 35 U.S.C. § 103(a) as being obvious over Chan in view of MSDN ("Lowering Total Cost of Ownership with Active Directory-Enabled Applications"). However, Applicant respectfully submits that the MSDN reference was cited for reasons unrelated to the utilization of an "aggregatable object" and thus fails to cure the defects of the Chan reference discussed earlier herein. Specifically—and without conceding the appropriateness of this rejection with regard to the other criteria for a finding of obviousness—Applicant respectfully submits that neither Chan nor MSDN, alone or in combination, teach the utilization of an "aggregatable object" as that term is defined by the present Application and, thus, this combination of references fails to teach or suggest each and every element present in Claim 14.

Therefore, for reasons substantially similar to those presented earlier herein with regard to Claims 1-13 and 15-29, Applicant respectfully submits the neither Chan nor MSDN, alone or in combination, teaches or suggests the utilization of an aggregatable object to extend a directory services interface. Applicant therefore requests that the rejection as to Claim 14 be withdrawn and that this claim also be allowed to issue.

**CONCLUSION**

Based on the reasons and rationale set forth herein, Applicant respectfully submits that the objections and rejections have been overcome and, accordingly, Applicant requests that the objections and rejections be withdrawn and that the pending claims be allowed to issue. Should the Examiner have any questions, comments, or suggestions that would expedite the prosecution of the present case to allowance, Applicant's undersigned representative earnestly requests a telephone conference at (206) 332-1394.

Date: February 8, 2005

  
Richard W. Knight  
Registration No. 42,751

Woodcock Washburn LLP  
One Liberty Place - 46th Floor  
Philadelphia PA 19103  
Telephone: (215) 568-3100  
Facsimile: (215) 568-3439